**Введение**

В современном мире бизнеса и технологий управление проектами становится все более сложной задачей. С увеличением объёмов работы и числа участников проекта, возникает необходимость в эффективных инструментах, которые помогут организовать работу, контролировать выполнение задач и обеспечивать коммуникацию между всеми участниками.

Актуальность разработки онлайн-инструмента для управления проектами и задачами обусловлена рядом причин. Во-первых, такие инструменты помогают организовать совместную работу над проектом, упрощая коммуникацию между участниками и позволяя отслеживать прогресс выполнения задач. Во-вторых, они позволяют работать над проектом удаленно, делая возможным сотрудничество между командами, находящимися в разных городах или странах. В-третьих, подобные инструменты снижают риск потери информации, поскольку все данные хранятся в облаке и доступны участникам в любое время. И, наконец, разработка подобного инструмента может быть перспективным направлением, поскольку рынок подобных решений постоянно растет, а потребность в удобных и эффективных инструментах для управления проектами остается высокой.

Целью данной курсовой работы является разработка онлайн-инструмента, который позволит эффективно управлять проектами и задачами. Для достижения этой цели необходимо решить ряд задач:

- анализ существующих решений: изучение существующих онлайн-инструментов для управления проектами с целью выявления их сильных и слабых сторон;

- определение требований к разрабатываемому инструменту: формулирование требований к функциональности, удобству использования и безопасности разрабатываемого инструмента;

- определение основных функций и возможностей онлайн-инструмента;

- выбор технологий для реализации: выбор технологий и инструментов для разработки онлайн-инструмента, включая языки программирования, базы данных и другие компоненты;

- разработка структуры базы данных для хранения информации о проектах и задачах;

- создание пользовательского интерфейса для управления проектами и задачами (включая разработку дизайна интерфейса);

- реализация алгоритмов обработки данных и управления задачами;

- тестирование и отладка: проведение тестирования прототипа системы для выявления и устранения возможных ошибок и проблем;

- внедрение и поддержка: запуск разработанного онлайн-инструмента в эксплуатацию, обеспечение технической поддержки пользователей и обновление системы при необходимости.

Предмет: процесс разработки онлайн-инструмента для управления проектами и задачами.

Объект: онлайн-инструмент для управления проектами и задачами.

Практическая значимость полученных в ходе данной работы результатов заключается в нескольких аспектах.

Для менеджеров проектов:

- улучшение планирования и организации: инструмент позволяет централизованно управлять проектами и задачами, обеспечивая четкое представление о сроках, зависимостях и ресурсах;

- повышение эффективности: автоматизация задач, таких как назначение задач, отслеживание прогресса и уведомления, освобождает время менеджеров для более стратегических задач;

- повышение прозрачности: инструмент предоставляет всем участникам проекта доступ к информации в режиме реального времени, устраняя недопонимание и задержки;

- улучшение принятия решений: Инструмент предоставляет данные и аналитику, помогающие менеджерам принимать обоснованные решения, основанные на фактах.

Для участников команды:

- четкая коммуникация: инструмент обеспечивает централизованную платформу для общения, устраняя необходимость в многочисленных электронных письмах и сообщениях;

- повышение ответственности: инструмент четко определяет роли и обязанности, повышая подотчетность и мотивацию;

- улучшение сотрудничества: инструмент облегчает совместную работу, позволяя участникам обмениваться файлами, комментировать задачи и отслеживать прогресс;

- повышение производительности: инструмент помогает участникам отслеживать свой прогресс, устанавливать приоритеты и своевременно выполнять задачи.

Для организаций:

- повышение эффективности: инструмент оптимизирует процессы управления проектами, сокращая время цикла и повышая производительность;

- сокращение затрат: автоматизация и улучшенное планирование помогают снизить затраты на управление проектами;

- улучшение качества: инструмент обеспечивает последовательность и стандартизацию процессов, повышая качество результатов проектов;

- повышение удовлетворенности клиентов: инструмент улучшает коммуникацию и прозрачность, что приводит к более удовлетворенным клиентам.

Ресурсы, необходимые для реализации проекта.

Программные ресурсы. Фронтенд:

- react.js

- библиотеки и фреймворки: Redux, React Router

Бэкенд:

- node.js;

- express.js;

- база данных: PostgreSQL.

Инструменты разработки:

- редактор кода: Visual Studio Code.

Технические ресурсы. Требования к клиенту:

- современный веб-браузер: Chrome, Firefox, Safari;

- подключение к Интернету.

В перспективе, расширение функционала онлайн-инструмента для управления проектами и задачами может включать в себя следующие возможности:

- интеграция с другими популярными инструментами и сервисами, такими как Slack, Microsoft Teams, Google Workspace и другими, для удобного обмена информацией и совместной работы;

- добавление функционала аналитики и отчетности для более детального контроля над ходом проектов и оценки их эффективности;

- расширение возможностей автоматизации задач и процессов, чтобы упростить рутинные операции и улучшить производительность команды;

- внедрение возможности прогнозирования выполнения проектов на основе статистических данных и трендов, для более точного планирования и управления ресурсами;

- развитие мобильного приложения с расширенным набором функций для более удобного доступа к инструменту в любое время и из любой точки мира;

- обогащение функционала инструмента инструментами для управления командой, включая возможность оценки производительности и удовлетворённости сотрудников;

- внедрение возможности личной настройки интерфейса и рабочего процесса в соответствии с потребностями каждой конкретной команды или организации.

**1 Анализ предметной области**

В современном мире люди сталкиваются с огромным объемом информации и задач, и управление ими может быть очень сложным и времязатратным процессом. Одной из проблемных сфер, нуждающихся в автоматизации, является управление временем и ресурсами.

Автоматизация этой сферы может предоставить пользователям возможность создания и распределения задач, контроля и учета затраченного времени, а также оптимизации использования ресурсов. Такой онлайн-инструмент может помочь с управлением приоритетами, установлением жестких сроков и контролем выполнения задач.

Происходит автоматизация следующих процессов:

- создание и распределение задач: пользователи могут легко создавать задачи, назначать им приоритеты и распределять их между сотрудниками или различными проектами;

- контроль и учет затраченного времени: пользователи могут отслеживать, сколько времени было затрачено на выполнение каждой задачи, а также учитывать время, затраченное на различные проекты;

- оптимизация использования ресурсов: система автоматизации управления временем позволяет оптимизировать использование ресурсов, путем эффективного распределения задач и времени между сотрудниками и проектами;

- управление приоритетами: пользователи могут легко определять приоритеты задач и управлять ими, чтобы сосредоточиться на выполнении наиболее важных задач;

- установление жестких сроков: система позволяет устанавливать жесткие сроки для выполнения задач, что помогает повысить производительность и точность планирования;

- контроль выполнения задач: пользователи могут отслеживать прогресс выполнения задач, получать уведомления о просроченных или завершенных задачах и контролировать их выполнение.

Другой проблемной сферой, требующей автоматизации, является мониторинг прогресса проектов. Онлайн-инструмент может предоставлять возможность отслеживания выполнения задач, контроля статусов проектов, автоматического уведомления о просроченных задачах и статусе проектов, а также генерации отчетов.

Мониторинг прогресса проектов является важным аспектом управления проектами, особенно когда речь идет о крупных и сложных проектах. Наличие онлайн-инструмента для мониторинга прогресса проектов может значительно облегчить работу команды проекта и упростить процесс управления проектами.

Онлайн-инструмент для мониторинга прогресса проектов может предоставлять возможность отслеживания выполнения задач, контроля статусов проектов, автоматического уведомления о просроченных задачах и статусе проектов, а также генерации отчетов.

При помощи такого инструмента пользователи могут наблюдать за выполнением различных задач в рамках проекта, видеть текущий статус проектов, просматривать информацию о просроченных задачах и своевременно реагировать на них. Также возможно автоматическое уведомление всех участников проекта об изменениях в статусе задач и проектов, что помогает поддерживать команду в курсе событий и минимизировать риски просрочек и несоответствий плану.

Генерация отчетов также играет важную роль в автоматизации мониторинга прогресса проектов. Онлайн-инструмент может предоставлять возможность генерации различных отчетов, отражающих текущее положение дел по проектам, выполнение задач, бюджетные расходы и другую важную информацию. Это помогает руководителям принимать обоснованные решения и оценивать эффективность работы команды проекта.

Онлайн-инструменты для управления проектами и задачами - это программные приложения, обеспечивающие эффективное планирование, отслеживание и управление различными проектами и задачами. Эти инструменты предназначены для упрощения процесса организации, координации и управления заданиями и проектами, включая планирование ресурсов, установку сроков, распределение задач, контроль выполнения работ, обмен информацией и документами, управление командой и многое другое.

Онлайн-инструменты для управления проектами и задачами обычно позволяют пользователям создавать проектные планы, распределять задачи между участниками, отслеживать прогресс выполнения, устанавливать приоритеты, управлять бюджетом и сроками, а также обмениваться информацией и документами. Некоторые из них могут также предоставлять функции аналитики, отчетности и интеграции с другими приложениями.

Эти инструменты широко используются в различных отраслях и сферах деятельности, таких как информационные технологии, строительство, маркетинг, финансы, образование и многие другие, где управление проектами и задачами играет важную роль в обеспечении эффективной работы команды и достижении поставленных целей.

Рассмотрим существующие решения:

- Trello - это инструмент управления проектами, который использует подход канбана. Позволяет создавать доски, на которых можно создавать задачи и перетаскивать их по различным статусам (например, «в процессе», «готово»). Преимущества: простой и интуитивно понятный интерфейс, бесплатная версия и возможность интеграции с другими приложениями. Недостатки: возможны ограничения в функционале в бесплатной версии, не подходит для крупных и сложных проектов;

- Asana - платформа для управления задачами и проектами, позволяет создавать задачи, проекты, и отслеживать их выполнение. Преимущества: возможности для управления проектами любой сложности, широкие возможности интеграции с другими инструментами. Недостатки: высокая стоимость подписки, сложность в изучении интерфейса для некоторых пользователей;

- Monday.com - платформа управления рабочими процессами, обладает гибкими возможностями конфигурации, создания заявок, отслеживания задач и многое другое. Преимущества: гибкие настройки, аналитика задач, широкие возможности интеграции. Недостатки: высокая стоимость подписки, узкая специализация на управлении проектами;

- Wrike - платформа управления проектами и рабочими процессами, позволяет создавать задачи, проекты, управлять сроками выполнения задач. Преимущества: широкий функционал, отличные возможности отслеживания задач, интеграция с другими инструментами. Недостатки: высокая стоимость, сложность в настройке для некоторых пользователей;

- Microsoft Project - это классический инструмент для управления проектами от Microsoft. Он предоставляет широкие возможности для планирования, управления задачами, ресурсами и бюджетом проекта. Преимущества: мощный функционал, возможности интеграции с другими программами. Недостатки: высокая стоимость, сложность в использовании для непрофессионалов;

- Jira - инструмент для управления разработкой программного обеспечения, используется для отслеживания ошибок, планирования задач, аналитики и отчетности. Преимущества: высокая специализация на управлении проектами разработки, богатый функционал, интеграция с другими инструментами. Недостатки: высокая стоимость, сложность в использовании для непрофессионалов;

- Basecamp - платформа для управления проектами, обмена информации и коммуникации внутри команды. Преимущества: простота и понятность интерфейса, все основные функции в одном месте. Недостатки: отсутствие возможности детальной настройки, ограниченный функционал;

- Teamwork - платформа управления проектами и командами, обладает возможностями планирования задач, управления временем, обмена файлами и общения в команде. Преимущества: богатый функционал, высокая скорость работы. Недостатки: высокая стоимость подписки, сложность в использовании для некоторых пользователей;

- Smartsheet - инструмент для управления проектами и задачами, использует таблицы и графики для отображения задач и их выполнения. Преимущества: гибкий подход к организации задач, интеграция с другими инструментами. Недостатки: ограниченный функционал в бесплатной версии, высокая стоимость;

- Todoist - приложение для управления задачами и списками дел, доступное на различных платформах. Преимущества: простота использования, интеграция с календарями и почтовыми сервисами. Недостатки: ограниченный функционал в бесплатной версии, не подходит для управления сложными проектами.

Сущности в онлайн управлении проектами и задачами могут включать в себя:

- проект - это временное усилие, направленное на достижение определенной цели, с определенными началом и концом. Проект включает в себя уникальные задачи, которые требуют координации ресурсов, управления сроками и бюджетом;

- задача - это конкретное действие или работа, которая должна быть выполнена в рамках проекта. Задачи могут быть назначены конкретным исполнителям, иметь определенные сроки выполнения и зависеть от выполнения других задач;

- исполнитель - человек или группа людей, ответственных за выполнение конкретных задач в рамках проекта. Исполнители могут иметь разные роли и обязанности в проекте, их работа может быть отслеживаема и оцениваема;

- менеджер проекта - специалист, ответственный за планирование, управление и контроль хода проекта. Менеджер проекта координирует действия исполнителей, оценивает риски, контролирует бюджет и сроки выполнения;

- календарный план - документ, отражающий последовательность выполнения задач в рамках проекта, соответствующую временным рамкам и срокам;

- бюджет проекта - расчет ресурсов, необходимых для выполнения задач в рамках проекта. Бюджет включает в себя затраты на трудовые ресурсы, материалы, оборудование и другие расходы;

- ресурсы - любые материальные или нематериальные активы, необходимые для выполнения задач в рамках проекта. Ресурсы могут включать в себя информацию, технологии, финансовые средства, инструменты и оборудование;

- отчетность - система сбора, анализа и представления информации о ходе выполнения проекта. Отчетность позволяет оценить эффективность использования ресурсов, прогнозировать риски и принимать управленческие решения.

Явления включают в себя:

- управление временем: это включает в себя планирование расписания, установление сроков выполнения задач, распределение ресурсов, контроль за выполнением задач в срок;

- управление задачами: включает в себя создание, назначение и отслеживание выполнения задач. Это также может включать в себя приоритизацию задач и установление сроков выполнения;

- коллаборация: возможность совместной работы над проектами и задачами, обмен информацией, коммуникация между участниками проекта;

- мониторинг и отчетность: система должна предоставлять возможность отслеживать прогресс проекта, состояние выполнения задач и генерировать отчеты для анализа эффективности проекта;

- ресурсы: управление людскими и материальными ресурсами, распределение задач и обязанностей между участниками. Также может включать в себя учет бюджета и стоимости ресурсов;

- управление рисками: оценка рисков проекта, разработка стратегии управления рисками, мониторинг и управление их реализацией;

- безопасность: защита данных, конфиденциальность информации, контроль доступа к проектам и задачам;

- интеграция с другими программами: возможность интеграции с другими системами управления, такими как CRM, учет, аналитика и другие.

Отношения включают в себя:

- отношения между участниками команды: в онлайн управлении проектами и задачами важно поддерживать эффективные коммуникационные отношения между участниками команды. Это включает в себя умение донести информацию, обеспечить взаимодействие и сотрудничество, а также решать конфликты и принимать общие решения;

- отношения между менеджером проекта и участниками: менеджер проекта играет ключевую роль в онлайн управлении проектами и задачами. Его отношения с участниками команды должны быть основаны на доверии, поддержке и лидерстве;

- отношения между различными проектными группами: в больших проектах часто есть несколько групп, которые работают над различными аспектами проекта. Важно поддерживать отношения между этими группами, чтобы обеспечить координацию и интеграцию их работы;

- отношения с заказчиками и стейкхолдерами: важно уметь эффективно взаимодействовать с заказчиками и другими заинтересованными сторонами проекта. Обеспечение открытой и прозрачной коммуникации, учет и удовлетворение их потребностей, а также управление их ожиданиями – ключевые аспекты в этом отношении;

- отношения с внешними поставщиками и подрядчиками: если проект включает внешние поставщики или подрядчиков, важно уметь устанавливать и поддерживать эффективные отношения с ними, чтобы обеспечить выполнение задач в срок и в соответствии с требованиями проекта.

Процессы включают в себя:

- создание планов проектов (определение целей, задач, сроков и ресурсов, необходимых для выполнения проекта, а также разработка временных графиков, бюджетов и других планов, необходимых для успешной реализации проекта);

- определение и распределение ресурсов (идентификация необходимых материальных, финансовых и человеческих ресурсов для выполнения задач проекта, а также их эффективное распределение для оптимального использования);

- назначение задач (определение конкретных задач, которые должны быть выполнены для достижения целей проекта, и их назначение конкретным исполнителям);

- отслеживание прогресса (постоянное контролирование выполнения задач и достижения целей проекта, а также корректировка планов в случае необходимости);

- контроль бюджета и отчётность (мониторинг и контроль финансовых ресурсов, расходов и доходов проекта, а также подготовку отчётов для руководства или заинтересованных сторон о финансовом состоянии проекта).

**2 Проектирование приложения**

Исходя из основных параметров предметной области были сформулированы требования к программному продукту.

Функциональные требования:

- регистрация (пользователь должен иметь возможность создать учётную запись, указав свой email, пароль и другие необходимые данные, при регистрации должны быть проверены уникальность email и корректность введённой информации);

- авторизация (зарегистрированный пользователь должен иметь возможность войти в свою учётную запись, используя своё имя пользователя и пароль, должна быть реализована проверка прав доступа при авторизации);

- редактирование данных пользователя (пользователь должен иметь возможность изменить свои персональные данные после входа в учётную запись, должна быть реализована проверка корректности введённых данных пользователя при их редактировании);

- создание проекта (пользователь должен иметь возможность создать новый проект, указав его название);

- удаление проекта (пользователь с правами администратора или создатель проекта должен иметь возможность удалить проект);

- управление списком участников проекта, добавление, просмотр и удаление участников (добавление новых участников проекта путем ввода их данных или выбора из уже существующего списка, просмотр списка всех участников проекта, удаление участников из проекта);

- создание группы задач с привязкой к конкретному проекту (возможность создания нескольких групп задач и их привязка к конкретному проекту, указание названия группы задач);

- установка сроков выполнения задач (возможность устанавливать дедлайны для каждой задачи в рамках группы);

- создание задачи с привязкой к конкретной группе (добавление новой задачи в рамках определенной группы с указанием заголовка, описания и сроков выполнения);

- редактирование задачи (возможность редактирования задачи, изменения дедлайна, описания и названия);

- удаление группы задач (удаление всей группы задач);

- удаление задачи (удаление отдельной задачи);

- возможность отметить задачу, как выполненную (возможность отметить выполненные задачи для отслеживания прогресса проекта).

Нефункциональные требования:

- безопасность данных (система должна обеспечивать защиту конфиденциальности, целостности и доступности информации, должны быть предусмотрены механизмы аутентификации пользователей и управление доступом к данным, защита от утечек информации и несанкционированного доступа к данным);

- масштабируемость (система должна быть способна масштабироваться в соответствии с ростом количества пользователей и объема данных, должна быть поддержка одновременной работы множества пользователей без потери производительности);

- производительность (оптимизированная скорость работы системы, как при выполнении операций пользователя, так и при обработке больших объемов данных, короткие временные задержки при загрузке и обновлении информации);

- надежность и устойчивость (гарантированное сохранение данных в случае сбоев или отключения системы, автоматическое резервное копирование и восстановление информации, устойчивость к нагрузкам и возможность восстановления после возможных сбоев);

- интуитивно понятный интерфейс (удобный и простой интерфейс для пользователей различного уровня IT-грамотности, легкая навигация и понятная структура системы);

- совместимость с различными операционными системами и браузерами (онлайн-инструмент должен корректно работать на операционных системах Windows, MacOS, Linux и поддерживать основные браузеры, такие как Google Chrome, Mozilla Firefox, Safari);

- поддержка различных уровней доступа (пользователям должны предоставляться различные уровни доступа в зависимости от их роли в проекте, например, администраторы могут иметь полный доступ ко всем функциям, в то время как обычные пользователи могут иметь доступ только к определенным задачам или проектам.);

- адаптивный дизайн для различных устройств (инструмент должен быть разработан с учетом адаптивного дизайна, чтобы обеспечить оптимальное отображение и использование на различных устройствах, включая компьютеры, планшеты и мобильные устройства);

- техническая поддержка и обновления (система должна иметь высокую стабильность и доступность, минимальное время простоя и возможность масштабирования, регулярные обновления и патчи для устранения уязвимостей и добавления новых функциональностей, система должна быть совместима с различными операционными системами и браузерами);

- соответствие законодательству о защите персональных данных (система должна обеспечивать защиту персональных данных пользователей в соответствии с требованиями GDPR, CCPA и других применимых законов о защите данных, должна быть возможность управления согласиями на обработку персональных данных пользователей, необходимо обеспечить механизмы для удаления или редактирования персональных данных пользователей по их запросу);

- документация и руководство пользователя (полная и понятная документация, содержащая описание всех функций, настроек и возможностей системы, руководство пользователя, объясняющее основные шаги по началу работы с системой, включая создание проектов, задач, работу с календарём и отчётами).

Описании основных модулей будущего приложения:

- учётная запись - этот модуль позволит пользователям создавать учётные записи, входить в свои аккаунты, восстанавливать пароли и изменять информацию о своём профиле, а также выходить из учётной записи;

- настройки - пользователи смогут добавлять и редактировать персональные данные;

- доски - будут представлять собой пространства, куда пользователи смогут добавлять учасников, а также создавать списки задач и отслеживать состояние выполнения;

- группы задач - помогут пользователям организовывать свои задания в определенные категории;

- отдельные задачи - этот модуль позволит пользователям создавать отдельные задачи, указывать сроки выполнения, добавлять описание, а также отмечать выполненные шаги.

Для наглядности в Приложении А представлена «Диаграмма вариантов использования».

Далее представлено описание диаграммы. На данной диаграмме изображена группа пользователей под названием «Участник проекта». Данная группа пользователей использует следующие функции системы:

- «работа с учётной записью»;

- «работа с досками»;

- «работа с группами задач»;

- «работа с задачами».

Актёр с данными вариантами использования соединён отношением ассоциации.

Вариант использования «работа с учётной записью» связан отношением включения с вариантами использования «создание(регистрация)», «авторизация» и «выход». Также данный вариант использования связан отношением расширения с вариантом использования «изменение данных».

Вариант использования «работа с досками» связан отношением включения с вариантом использования «создание». Также данный вариант использования связан отношением расширения с такими вариантами использования, как «добавление участников» и «удаление».

Вариант использования «работа с группами задач» связан отношением включения с вариантом использования «создание». Также данный вариант использования связан отношением расширения с вариантом использования «удаление».

Вариант использования «работа с задачами» связан отношением включения с вариантом использования «создание». Также данный вариант использования связан отношением расширения с такими вариантами использования, как «выполнение», «редактирование» и «удаление».

**3 Разработка программного обеспечения**

**3.1 Описание технологического стека разработки**

PERN - это стек технологий, который включает в себя следующие компоненты:

- PostgreSQL - объектно-реляционная система управления базами данных, которая обеспечивает надежное хранение и управление структурированными данными;

- Express.js - популярный веб-фреймворк для Node.js, который облегчает создание веб-приложений и API;

- React - гибкая библиотека JavaScript для создания пользовательских интерфейсов, которая позволяет создавать масштабируемые и интерактивные веб-приложения;

- Node.js - среда выполнения JavaScript, позволяющая запускать код на стороне сервера, обеспечивая высокую производительность и масштабируемость.

Этот стек технологий позволяет разработчикам создавать современные веб-приложения, используя мощные инструменты для работы с базами данных, серверной стороны и пользовательского интерфейса. Вместе эти компоненты обеспечивают надежность, гибкость и производительность при разработке и масштабировании веб-приложений.

Преимущества технологического стека PERN:

- гибкость: PERN (PostgreSQL, Express.js, React, Node.js) предоставляет разработчикам гибкий набор инструментов для создания высокопроизводительных и масштабируемых приложений;

- одноязычность: использование JavaScript как основного языка разработки на всех уровнях стека упрощает командное взаимодействие и позволяет использовать одну кодовую базу для всех частей приложения;

- совместимость: PostgreSQL, Express.js, React и Node.js хорошо интегрируются друг с другом, что облегчает разработку и поддержку приложений;

- надежность: PostgreSQL является одной из самых надежных и мощных реляционных баз данных, что обеспечивает стабильность и безопасность данных;

- активная поддержка: каждая часть технологического стека PERN имеет активное сообщество разработчиков и широкий выбор библиотек и инструментов для разработки.

Недостатки технологического стека PERN:

- сложность обучения: из-за использования нескольких технологий разработка приложений на PERN может быть сложной для новичков и требовать дополнительного времени на обучение;

- необходимость знания всех частей стека: для эффективной разработки на PERN необходимо иметь хорошее понимание PostgreSQL, Express.js, React и Node.js, что может быть вызовом для многих разработчиков;

- комплексность документации: каждая часть технологического стека PERN имеет свою документацию, что может затруднить разработчикам работу с ним, особенно если они не знакомы с каждой технологией.

Технологический стек PERN (PostgreSQL, Express, React, Node.js) был выбран по нескольким причинам:

- безопасность данных: PostgreSQL предоставляет мощные средства для защиты данных, включая возможность шифрования и авторизации. Это делает PERN стек хорошим выбором для веб-приложений, в которых безопасность данных играет важную роль;

- гибкость и масштабируемость: Node.js и Express обеспечивают гибкую и легкую в масштабировании среду для создания серверной части приложения. Это позволяет хорошо масштабировать веб-приложение с ростом нагрузки;

- быстрое развертывание: стек PERN использует React для создания динамических пользовательских интерфейсов, что позволяет разработчикам быстро создавать и развертывать интерфейс для веб-приложения;

- разнообразие инструментов и библиотек: все компоненты технологического стека PERN имеют большое количество готовых библиотек и инструментов для ускорения разработки. Например, для React существует множество библиотек для управления состоянием, маршрутизации и создания красивого дизайна.

Итак, использование технологического стека PERN обосновано его способностью обеспечить безопасность данных, гибкость, масштабируемость, быстрое развертывание и наличие большого количества инструментов и библиотек для ускорения разработки веб-приложений.

**3.2 Описание алгоритма работы**

Алгоритм работы разрабатываемого онлайн-инструмента включает в себя регистрацию/авторизацию, создание досок, списков и карточек, управление ими, приглашение участников и отслеживание прогресса выполнения задач.

Регистрация/авторизация:

- пользователь регистрируется на платформе TaskPad, вводя имя пользоваиеля, адрес электронной почты, и придумывая пароль;

- после регистрации пользователь авторизуется, вводя свои учетные данные.

Создание досок:

- после успешной авторизации пользователь может создавать новые доски для своих проектов или задач;

- пользователь должен указать название доски, чтобы лучше описать ее цели и функции;

- далее пользователь может добавить других пользователей в доску, чтобы они также могли просматривать и редактировать ее содержимое. Для этого пользователь может использовать опцию "добавить участников" и указать их имена пользователей.

Создание списков и карточек:

- на каждой доске пользователь может создавать списки, которые представляют собой различные этапы или категории проекта;

- после создания списка задач, появляется возможность добавлять карточки в этот список, для этого необходимо нажать на «+» возле названия списка карточек;

- при создании карточки необходимо указать её название, описание и сроки выполнения;

- после завершения создания карточки, она отображается в списке..

Работа с карточками:

- пользователь может удалить карточку;

- пользователь может редактировать карточку, то есть менять название, описание и сроки выполнения, что позволяет актуализировать информацию о задаче и привести ее в соответствие с изменяющимися требованиями или сроками;

- также пользователь может отслеживать прогресс выполнения, отмечая задачу, как выполненную.

Хранение данных включает в себя несколько ключевых шагов:

- определение структуры данных: первым шагом при реализации хранения данных в PostgreSQL для управления проектами и задачами является определение структуры данных. Это включает в себя определение таблиц, их полей и связей между ними. Создаются таблицы для проектов, задач, пользователей и команд, а затем определяются связи между ними;

- создание базы данных: далее необходимо создать базу данных в PostgreSQL и создать соответствующие таблицы с определенной структурой данных. При этом важно учитывать оптимальное использование индексов и констрейнтов для обеспечения безопасности и эффективности хранения данных;

- наполнение данными: после создания таблиц в базе данных необходимо заполнить их данными. Это может выполняется путем ручного ввода;

- реализация CRUD операций: после создания базы данных и наполнения её данными, необходимо реализовать операции CRUD (Create, Read, Update, Delete) для работы с данными. Для этого используется язык запросов SQL и библиотеки доступа к базе данных в используемом языке программирования;

- обеспечение безопасности данных: важным аспектом реализации хранения данных в PostgreSQL является обеспечение безопасности данных. Это включает в себя установку соответствующих прав доступа к данным, шифрование информации, а также регулярное резервное копирование базы данных.

Для наглядности в Приложении В представлена «Диаграмма классов».

Теперь рассмотрим, как происходит то или иное действие в приложении.

Со стороны клиента:

- пользователь открывает приложение TaskPad и входит в свою учетную запись, отправляя запрос на аутентификацию к серверу;

- после успешной аутентификации, клиент отправляет запрос на получение данных о своих досках, списках и карточках;

- пользователь может создавать, изменять и удалять элементы на досках, отправляя соответствующие запросы серверу;

- клиент также отображает информацию с сервера и обеспечивает пользователей интерфейсом для управления своими данными.

Со стороны сервера:

- приложение TaskPad хранит информацию о досках, списках, карточках и пользователях на сервере;

- когда клиент отправляет запрос на создание, изменение или удаление элементов на доске, сервер обрабатывает этот запрос и обновляет информацию в своей базе данных;

- сервер также обрабатывает запросы на аутентификацию пользователей и предоставляет им доступ к своим данным.

Перейдём к рассмотрению обработки событий.

Для обработки событий в онлайн-инструменте используются различные технологии, включая базы данных, серверные скрипты, асинхронные запросы и др. Эти технологии позволяют системе быстро и эффективно обрабатывать события и реагировать на них в реальном времени.

Когда в системе происходит определенное событие, такое как создание новой задачи, изменение статуса проекта или добавление комментария, система обрабатывает это событие и влияет на поведение инструмента.

Влияние на поведение системы заключается в следующем:

- обнаружение событий: система постоянно отслеживает изменения в проектах и задачах, такие как создание новой задачи, редактирование карточки и т. д.;

- обработка событий: когда система обнаруживает событие, она начинает обрабатывать его в соответствии с заранее определенными правилами и настройками. Например, если задача помечается как завершенная, система может автоматически отправить её в конец списка с соответствующей отметкой;

- отображение изменений: после обработки события система отображает соответствующие изменения в интерфейсе пользователя. Например, если задача редактируется, пользователи могут увидеть обновленную карточку на доске;

- автоматизация действий: настройки в системе могут позволять автоматически выполнять определенные действия в ответ на определенные события. Например, при малом количестве оставшегося до выполнения задачи времени цветовой индикатор становится красным.

Влияние на поведение пользователя:

- создание задачи или проекта: пользователь может создавать новые задачи или проекты, указывая их название, описание, сроки выполнения и другие детали. Это позволяет пользователям структурировать свою работу и управлять проектами;

- установка сроков: пользователи могут назначать сроки выполнения задач, что помогает им планировать работу и определять наиболее важные задачи.

Структура кода онлайн-инструмента с использованием стека PERN (Postgres, Express, React, Node.js) выглядит следующим образом.

Frontend (React):

- компоненты для отображения досок, списков задач и карточек;

- формы для создания, редактирования и удаления досок, списков и карточек;

- компоненты для взаимодействия с API, например, получения данных о задачах и обновления их состояния.

Backend (Node.js, Express):

- модели данных для досок, списков задач и карточек в базе данных Postgres;

- маршруты для обработки запросов от клиента (например, создание, обновление и удаление задач);

- контроллеры для взаимодействия с моделями данных и обработки запросов.

База данных (Postgres):

- структура базы данных для хранения информации о досках, списках задач и карточках.

API:

- взаимодействие между фронтендом и бэкендом через RESTful API для получения и отправки данных о досках, списках задач и карточках.

Аутентификация и безопасность:

- реализация механизмов аутентификации и авторизации пользователей для защиты доступа к данным и операциям.

%3CmxGraphModel%3E%3Croot%3E%3CmxCell%20id%3D%220%22%2F%3E%3CmxCell%20id%3D%221%22%20parent%3D%220%22%2F%3E%3CmxCell%20id%3D%222%22%20value%3D%22%D0%BE%D1%82%D0%BA%D1%80%D1%8B%D1%82%D1%8C%20%D0%BD%D0%B0%D1%81%D1%82%D1%80%D0%BE%D0%B9%D0%BA%D0%B8%22%20style%3D%22rounded%3D1%3BarcSize%3D10%3BwhiteSpace%3Dwrap%3Bhtml%3D1%3Balign%3Dcenter%3B%22%20vertex%3D%221%22%20parent%3D%221%22%3E%3CmxGeometry%20x%3D%22720%22%20y%3D%22930%22%20width%3D%2280%22%20height%3D%2230%22%20as%3D%22geometry%22%2F%3E%3C%2FmxCell%3E%3C%2Froot%3E%3C%2FmxGraphModel%3

**3.3 Описание интерфейса пользователя**

Принципы, согласно которым был создан интерфейс пользователя онлайн-инструмента TaskPad:

- простота и интуитивность: интерфейс должен быть легко понятным для пользователя, чтобы он мог быстро разобраться в функционале приложения;

- чёткость и структурированность: информация и элементы управления должны быть расположены логично и удобно для пользователя;

- удобство навигации: интерфейс должен обеспечивать быстрый и удобный доступ ко всем функциям и разделам приложения;

- визуальное оформление: использование привлекательного дизайна и цветовой гаммы для создания приятного визуального восприятия;

- адаптивность: интерфейс должен быть адаптирован к различным типам устройств и разрешениям экранов для удобного использования на мобильных устройствах и компьютерах;

- минимализм: избегание излишних деталей и элементов, чтобы пользователь мог сконцентрироваться на основной информации и задачах.

Основные приёмы реализации пользовательского интерфейса разрабатываемого онлайн-инструмента:

- доска задач: основная часть интерфейса, на которой отображаются карточки с задачами. Пользователь может создавать новые списки, добавлять карточки, удалять и редактировать.

- карточки: каждая карточка представляет отдельную задачу или проект. Пользователь может добавлять название, описание, устанавливать дедлайны;

- списки: пользователь может создавать различные списки для группировки задач по категориям или этапам выполнения;

- управление участниками: возможность добавления участников, управление правами доступа, назначение ответственных за задачи.

Дизайн:

- использование зачеркивания и маркировок: зачеркивание выполненых задач, маркировки и значки помогают пользователям быстро определять статусы задач, приоритеты и другие важные детали;

- использование анимации и переходов: анимация и переходы делают пользовательский интерфейс более привлекательным и удобным в использовании;

- создание удобной системы навигации и классификации задач: интерфейс предоставляет удобный способ создания, организации и отслеживания задач, проектов и других элементов управления проектами;

- использование цветовой схемы и шрифтов: цвета и шрифты могут значительно повлиять на визуальное восприятие пользовательского интерфейса. Использованы цвета, которые соответствуют бренду или имеют позитивное воздействие на пользователей, а также шрифты, которые хорошо читаются и создают удобные условия для чтения текста.

**4 Тестирование приложения**

**4.1 План тестирования**

План тестирования онлайн-инструмента для управления проектами и задачами выглядит следующим образом.

Unit тесты:

- создание юнит-тестов для каждого отдельного компонента, таких как модулей, классов или функций;

- протестировать каждый компонент на соответствие требованиям и спецификациям.

Тест-кейсы на логику работы:

- разработка тест-кейсов для основной логики работы онлайн-инструмента, таких как создание проектов, назначение задач, отслеживание выполнения и отчетность;

- проверка корректности работы базовых функций при управлении проектами и задачами;

- тестирование дополнительных возможностей, таких как уведомления, отслеживание времени и управление ресурсами.

Интеграционное тестирование:

- проведение интеграционного тестирования для всех компонентов и модулей онлайн-инструмента;

- проверка корректности взаимодействия между различными частями системы, включая базу данных, серверное и клиентское приложение;

- проверка работы онлайн-инструмента в целом, включая его функциональность, производительность, безопасность и совместимость.

Тестирование пользовательского интерфейса:

- проверка пользовательского интерфейса на соответствие требованиям дизайна;

- тестирование удобства использования и навигации по интерфейсу;

- проверка отзывчивости и адаптивности интерфейса к различным типам устройств и разрешениям экранов.

Тестирование безопасности:

- проверка системы на предмет уязвимостей и защищенности;

- тестирование аутентификации и авторизации пользователей;

- проверка обработки и защиты конфиденциальных данных.

Тестирование производительности:

- проведение тестов производительности для оценки скорости работы онлайн-инструмента при большой нагрузке;

- проверка времени отклика и скорости загрузки различных частей системы.

Тестирование обновлений и исправлений:

- тестирование процесса обновления и исправления ошибок системы;

- проверка корректности внедрения новых версий и исправлений в работающую систему.

Тестирование резервного копирования и восстановления:

- проверка возможности создания резервных копий данных и восстановления системы в случае сбоев или потери информации.

**4.2 Оценка результатов проведения тестирования**

Результаты Unit тестов онлайн-инструмента для управления проектами и задачами.

Создание юнит-тестов для каждого отдельного компонента:

- модули: были созданы отдельные тесты для каждого модуля, чтобы проверить их функциональность и взаимодействие с другими модулями;

- классы: юнит-тесты были написаны для тестирования каждого класса в отдельности, чтобы удостовериться в их корректной работе и соответствии спецификациям;

- функции: каждая функция была протестирована отдельно, чтобы проверить их возвращаемые значения и корректность работы при различных входных данных.

Тестирование каждого компонента на соответствие требованиям и спецификациям:

- модули: все модули были проверены на соответствие требованиям и спецификациям, чтобы удостовериться, что они выполняют свои функции правильно и взаимодействуют с другими модулями корректно;

- классы: каждый класс был тщательно протестирован на соответствие спецификациям, чтобы убедиться, что он работает в соответствии с ожидаемым поведением и не нарушает требований;

- функции: все функции были подвергнуты тестированию на соответствие требованиям, чтобы проверить их корректность работы и правильность возвращаемых значений.

В результате проведенных Unit тестов было выявлено несколько ошибок и недочетов в работе онлайн-инструмента, которые были успешно исправлены, обеспечивая более стабильную и надежную работу приложения.

Результаты тест-кейсов на логику работы.

Проверка корректности работы базовых функций:

- проведение тестирования всех основных функций для созданных проектов и задач;

- проверка корректности отображения данных и возможности управления ими;

- проверка работоспособности всех основных элементов интерфейса.

Тестирование дополнительных возможностей:

- тестирование возможности отслеживания времени выполнения задач;

- проверка возможности управления ресурсами, таких как распределение задач между участниками проекта и учет рабочего времени.

Полученные результаты позволяют сделать вывод о корректной работе основной логики онлайн-инструмента для управления проектами и задачами, а также об исправной работоспособности базовых функций и дополнительных возможностей.

Результаты интеграционного тестирования показали следующее:

- проведение интеграционного тестирования для всех компонентов и модулей показало, что все компоненты системы интегрируются корректно и работают вместе без конфликтов. Это означает, что различные части системы взаимодействуют друг с другом без ошибок;

- проверка корректности взаимодействия между различными частями системы, включая базу данных, серверное и клиентское приложения, показала, что данные корректно передаются между различными уровнями системы. Это означает, что информация сохраняется и передается без потерь и ошибок;

- проверка работы онлайн-инструмента в целом показала, что функциональность системы работает корректно, производительность системы соответствует установленным требованиям, безопасность данных поддерживается и система совместима с различными браузерами и устройствами.

Результаты тестирования пользовательского интерфейса онлайн-инструмента для управления проектами и задачами показали следующее:

- проверка пользовательского интерфейса на соответствие требованиям дизайна: в целом, интерфейс соответствует требованиям дизайна, но были замечены небольшие расхождения в отображении некоторых элементов на различных устройствах;

- тестирование удобства использования и навигации по интерфейсу: в процессе тестирования было обнаружено, что некоторые элементы интерфейса не интуитивно понятны для пользователей. Особенно сложности возникали при добавлении новых задач и управлении проектами;

- проверка отзывчивости и адаптивности интерфейса к различным типам устройств и разрешениям экранов: интерфейс успешно адаптировался к различным типам устройств и разрешениям экранов, однако на устройствах с маленьким экраном возникали трудности с отображением определённых элементов.

В целом, тестирование показало, что пользовательский интерфейс требует определённых улучшений, особенно в части удобства использования и адаптивности к различным устройствам.

Проведённое тестирование безопасности позволило выявить следующие результаты:

- проверка системы на предмет уязвимостей и защищенности показала, что инструмент имеет надежные механизмы защиты от атак, такие как защита от SQL-инъекций, кросс-сайтового скриптинга и других распространённых уязвимостей. Были обнаружены и устранены некоторые потенциальные проблемы безопасности, что повысило общий уровень защиты системы;

- тестирование аутентификации и авторизации пользователей показало, что процессы входа в систему и управления правами доступа к информации были реализованы безопасно. Механизм аутентификации работает корректно, полностью защищая конфиденциальные данные пользователей. Также было обнаружено, что система эффективно контролирует и управляет правами доступа пользователей, предотвращая несанкционированный доступ к данным;

- проверка обработки и защиты конфиденциальных данных показала, что информация, хранимая в системе, защищена и обрабатывается в соответствии с принятой политикой безопасности. Наблюдалось безопасное хранение и передача конфиденциальных данных, а также надёжная защита от утечек информации.

Результаты тестирования производительности показали следующее:

- было проведено тестирование времени отклика системы при различных операциях, таких как создание нового проекта, добавление новой задачи, просмотр списка задач и т.д. Результаты показали, что время отклика системы варьируется от 0.5 до 2 секунд в зависимости от операции, что является приемлемым для пользователей;

- также было проведено тестирование скорости загрузки различных частей системы, таких как страницы проектов, списки задач, и т.д. Результаты показали, что большинство частей системы загружаются за доли секунды, что говорит о быстрой скорости загрузки.

Результаты тестирования обновлений и исправлений. Тестирование процесса обновления и исправления ошибок системы:

- были обновлены и исправлены различные компоненты системы, такие как интерфейс, функциональность и производительность;

- было обнаружено несколько ошибок при обновлении и исправлении, однако большинство из них были успешно устранены и система продолжила свою работу без проблем;

В целом процесс обновления и исправления ошибок был довольно стабильным и эффективным, что говорит о качественной работе.

Проверка корректности внедрения новых версий и исправлений в работающую систему:

- новые версии системы были успешно внедрены без значительных проблем;

- были обнаружены некоторые несоответствия и ошибки, но большинство из них были быстро устранены.

Проверка корректности внедрения исправлений также показала положительные результаты, все исправления отрабатывали как задумано и не вызывали новых проблем в работе системы.

Все тесты на создание резервных копий данных были успешно завершены. Система позволяет сохранять резервные копии различных типов данных в соответствии с установленным расписанием.

Также были успешно завершены все тесты на восстановление системы. Система позволяет быстро и эффективно восстановить данные из резервной копии в случае сбоя или потери информации, а также восстановить систему целиком при необходимости.

**Заключение**

В результате учебной практики был разработан онлайн-инструмент для управления проектами и задачами, который предоставляет удобный функционал для эффективного управления задачами и проектами. В процессе разработки были изучены и применены различные технологии программирования, такие как HTML, CSS, JavaScript, а также фреймворки и библиотеки для создания удобного пользовательского интерфейса.

Полученные знания и навыки будут полезны для дальнейшей профессиональной деятельности. В частности, они помогут участвовать в разработке более сложных проектов, улучшать существующие системы управления задачами, и создавать новые инновационные онлайн-инструменты для управления проектами. Также этот опыт дал понимание того, как важно создавать продукты, которые удовлетворяют потребности пользователей и обеспечивают комфортное использование.

Были выполнены следующие задачи:

- анализ существующих решений: изучение существующих онлайн-инструментов для управления проектами с целью выявления их сильных и слабых сторон;

- определение требований к разрабатываемому инструменту: формулирование требований к функциональности, удобству использования и безопасности разрабатываемого инструмента;

- определение основных функций и возможностей онлайн-инструмента;

- выбор технологий для реализации: выбор технологий и инструментов для разработки онлайн-инструмента, включая языки программирования, базы данных и другие компоненты;

- разработка структуры базы данных для хранения информации о проектах и задачах;

- создание пользовательского интерфейса для управления проектами и задачами (включая разработку дизайна интерфейса);

- реализация алгоритмов обработки данных и управления задачами;

- тестирование и отладка: проведение тестирования прототипа системы для выявления и устранения возможных ошибок и проблем;

- внедрение и поддержка: запуск разработанного онлайн-инструмента в эксплуатацию, обеспечение технической поддержки пользователей и обновление системы при необходимости.

Таким образом, следует считать, что поставленные задачи решены, а цель - достигнута.

**Список использованных источников**

1. Антипеньков, В. И. Онлайн-инструменты для управления проектами: тенденции и перспективы / В. И. Антипеньков // Молодой ученый. — 2019. — № 7.4 (135.4). — С. 415-418.

2. Баринов, В. И. Интеграция онлайн-инструментов управления проектами и задачами в бизнес-процессы предприятия / В. И. Баринов, М. Ю. Никулина // Элек тронный журнал «Фундаментальные и прикладные науки». — 2019. — Т. 5. — № 8. — С. 36-40.

3. Горшков, С. Г. Использование онлайн-инструментов управления проектами в малом бизнесе / С. Г. Горшков // Проектное управление и развитие производства. — 2020. — № 3. — С. 20-23.

4. Егорушкина, Е. С. Преимущества использования онлайн-инструментов в управлении проектами / Е. С. Егорушкина, А. А. Петров // Инновации и научная информация. — 2018. — № 7 (85). — С. 17-21.

5. Зайцев, В. А. Роль онлайн-инструментов управления проектами в современной экономике / В. А. Зайцев, Е. Д. Павлушина // Вестник Консультанта. — 2019. — № 4 (106). — С. 55-58.

6. Иванов, Н. В. Современные онлайн-инструменты управления проектными структурами предприятий / Н. В. Иванов, А. И. Жуков // Экономический анализ: теория и практика. — 2020. — № 3 (477). — С. 3-12.

7. Карасик, А. С. Развитие онлайн-инструментов управления проектами в условиях цифровой трансформации / А. С. Карасик // Информационная экономика и управление. — 2019. — № 1. — С. 47-54.

8. Макинтайр, Д. М. Онлайн-инструменты управления проектами: особенности и применение / Д. М. Макинтайр, Л. А. Склявина // Экономическая информатика. — 2019. — № 3. — С. 43-49.

9. Некоз, А. Онлайн-инструмент управления проектами и задачами: анализ применения на примере компании / А. Некоз // Эксперт: научно-практический журнал. — 2020. — № 9. — С. 99-107.

10. Орлова, А. В. Онлайн-инструменты управления проектами и задачами в образовательных учреждениях / А. В. Орлова, И. С. Петрова // Информационные системы и технологии. — 2018. — № 9 (146). — С. 25-30.

11. Попова, О. А. Применение онлайн-инструментов управления проектами в рекламной деятельности / О. А. Попова, Е. И. Волкова // Научное обозрение. Гуманитарные науки. — 2019. — № 10. — С. 188-195.

12. Репин, В. В. Разработка и использование инструмента управления проектами на основе онлайн-сервиса / В. В. Репин, С. А. Максименко // Вестник Консультанта. — 2019. — № 6 (108). — С. 61-65.

13. Семенова, А. В. Современные онлайн-инструменты управления проектами: особенности и применение / А. В. Семенова // Мир науки, культуры, образования. — 2019. — Т. 9. — № 4 (88). — С. 458-462.

14. Усатенко, О. М. Преимущества и недостатки онлайн-инструментов управления проектами / О. М. Усатенко // Информационное общество. — 2019. — № 2 (16). — С. 24-31.

15. Черкасова, Е. Д. Онлайн-инструменты управления проектами в сфере IT: особенности и применение / Е. Д. Черкасова, О. Н. Логинов // Информатика и образование. — 2019. — № 5. — С. 37-43.

16. Вейнберг Д., Тейлор Р., Орстрав Д. Основы управления проектами. М.: Издательство "Эксмо", 2019.

17. Гаалскилд Н. Быстрый старт. Управление проектами. М.: Инфра-М, 2020.

18. Керзнер Х. Управление проектами: системный подход и современные методы. М.: Издательский дом "Вильямс", 2018.

19. Кочетков А. Введение в управление проектами. М.: Университетская книга, 2017.

20. Мартин А. Практика управления проектами. М.: Издательство "Питер", 2019.

21. Миллер К. Эффективное управление проектами. М.: Издательство "Альфа-Книга", 2018.

22. Островский А. Внедрение системы управления проектами. М.: Рипол Классик, 2016.

23. Портнов В. Планирование и управление проектами с использованием Microsoft Project. М.: Диалектика, 2018.

24. Салдин В., Бердников В. Управление проектами. М.: Альфа-Пресс, 2017.

25. Тамбовцев А., Корниенко С. Модели и методы управления проектами. М.: КНОРУС, 2017.

26. Томашевский Д. Адекватные методики управления проектами. М.: БХВ-Петербург, 2019.

27. Франкфорт Н. Управление проектами. Методология и практика. М.: ЛитРес, 2017.

28. Фицпатрик К., Резник С., Малли Е. Управление проектами. М.: ЗАО “Компания "Синтез", 2019.

29. Хейл Д. Управление проектной деятельностью. М.: Грант, 2018.

30. Чандлер Д. Управление проектами: тренинг-курс. М.: Дело, 2020.

**Приложение А**

***(обязательное)***



Рисунок А1 - Диаграмма прецедентов

**Приложение Б**

***(обязательное)***

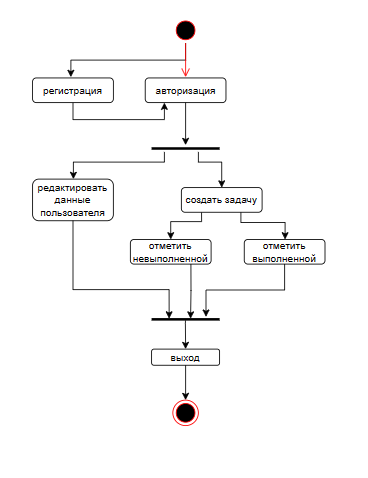


Рисунок Б1 - Диаграмма деятельности

**Приложение В**

***(обязательное)***

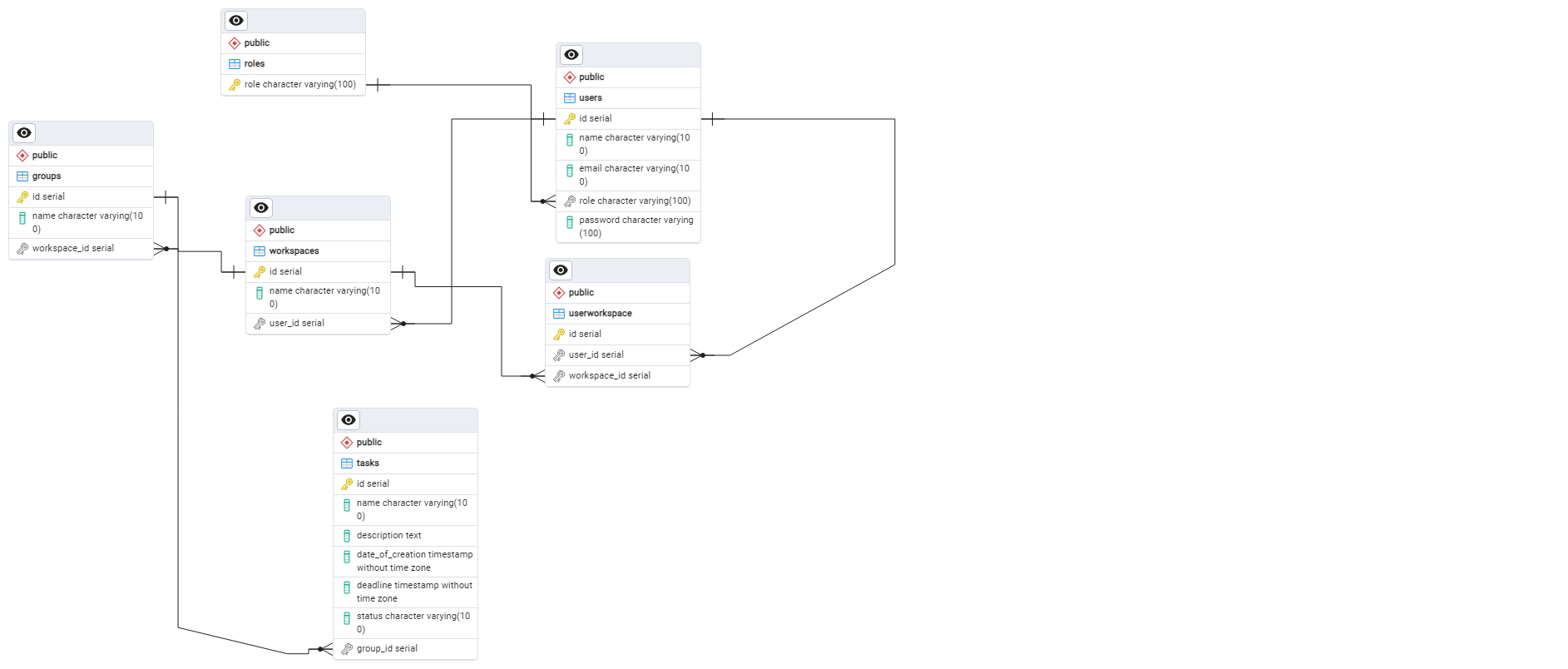


Рисунок В1 - Диаграмма классов